

Міністерство освіти і науки України  
Національний університет водного господарства та  
природокористування  
Кафедра комп'ютерних наук та прикладної математики

**04-01-48**

## **МЕТОДИЧНІ ВКАЗІВКИ**

**до курсової роботи з навчальної дисципліни**

**«Програмування»**

для здобувачів вищої освіти першого (бакалаврського) рівня за

спеціальностями 113 «Прикладна математика»,

121 «Інженерія програмного забезпечення»,

122 «Комп'ютерні науки»

денної, заочної та дистанційної форм навчання

Схвалено науково-методичною  
радою з якості ННІАКОТ  
Протокол № 7  
Від 27.03.2020 року

Рівне – 2020

Методичні вказівки до курсової роботи з навчальної дисципліни «Програмування» для здобувачів вищої освіти першого (бакалаврського) рівня за спеціальностями 113 «Прикладна математика», 121 «Інженерія програмного забезпечення», 122 «Комп'ютерні науки» денної, заочної та дистанційної форм навчання [Електронне видання] / Жуковський В. В., Прищеп О. В., Роценюк А. М., Ярошак С. В. – Рівне : НУВГП, 2020. – 15 с.

#### **Укладачі:**

Жуковський В. В. - к.т.н, доцент кафедри комп'ютерних наук та прикладної математики;

Прищеп О. В. - к.ф.-м.н, доцент кафедри комп'ютерних наук та прикладної математики;

Роценюк А. М. - к.п.н, старший викладач кафедри комп'ютерних наук та прикладної математики;

Ярошак С. В. - к.т.н, доцент кафедри комп'ютерних наук та прикладної математики.

#### **Відповідальний за випуск:**

Мартинюк П. М., д.т.н., професор, завідувач кафедри комп'ютерних наук та прикладної математики.

#### **Керівник групи забезпечення:**

113, Прищеп О. В., к.ф.-м.н, доцент кафедри комп'ютерних наук та прикладної математики;

121, Жуковська Н. А., к.т.н, доцент кафедри комп'ютерних наук та прикладної математики;

122, Мартинюк П. М., д.т.н., професор, завідувач кафедри комп'ютерних наук та прикладної математики.

© Жуковський В. В., Прищеп О. В.,  
Роценюк А. М., Ярошак С. В., 2020  
© НУВГП, 2020

## З М І С Т

1. ВСТУП.....	4
2. Загальні вимоги до курсової роботи.....	4
2.1. Курсова робота повинна містити	
2.2. Вимоги до розробки програми	
2.3. Основні етапи виконання курсової роботи	
3. Захист курсової роботи.....	5
3.1. Оформлення роботи	
3.2. Методика оцінювання курсових робіт	
3.3. Оцінки за захист курсової роботи	
4. Орієнтовний перелік тем курсової роботи з дисципліни «Програмування».....	8
4.1. Жуковський В. В. - 121 «Інженерія програмного забезпечення»	
4.2. Прищепя О. В. - 113 «Прикладна математика»	
4.3. Ярошак С. В. - 122 «Комп'ютерні науки»	
5. Приклад оформлення бібліографічного опису .....	12
5.1. Підручника, навчального посібника (один, два або більше авторів)	
5.2. Збірника наукових праць	
5.3. Статті з журналу	
5.4. Нормативних документів	
6. Рекомендована література.....	13
6.1. Основна література	
6.2. Електронні ресурси	
ДОДАТОК.....	14

## **1. ВСТУП**

Курсове проектування – одна з найважливіших частин навчального процесу, яка завершує вивчення більшості спеціальних дисциплін для поглиблення, закріплення й систематизації навичок самостійного, творчого підходу до розв’язування проблем ринку, удосконалення умінь, набутих під час практичних занять.

Курсова робота – це самостійна робота, в якій студент розробляє прогресивні рішення, використовуючи наукові принципи і методи, стандарти, нормативні документи.

Тематика курсової роботи відповідає вимогам нормативної програми і базується на техніко-економічних показниках реального світу.

**Завданням на курсову роботу** є створення програми згідно вибраної теми (див. п. 4). Перелік тем наведено нижче. Теми вибираються студентами та затверджуються викладачем. Крім того студент повинен оформити пояснювальну записку та підготувати презентацію.

## **2. ЗАГАЛЬНІ ВИМОГИ ДО КУРСОВОЇ РОБОТИ**

### **2.1. Курсова робота повинна містити**

1. Титульний лист (див. додаток).
2. Зміст (з вказанням сторінок).
3. Вступ (короткий виклад важливості розв’язуваного класу задач та доцільність її реалізації на ПК, опис предметної області).
4. Змістовна постановка задачі (приведення загального вигляду вхідних документів та вихідних форм, аналіз поставленої задачі, технічне завдання, опис інтерфейсу, діаграма варіантів використання).
5. Алгоритм розв’язання задачі (словесний або у вигляді блок-схеми, додаткові UML діаграми).
6. Текст програми (написаний від руки або надрукований).
7. Зображення та описання всіх форм, які є в проєкті.
8. Тестовий приклад (з наведеною повною вхідною інформацією, яка дозволяє перевірити всі режими роботи програми, всіма вихідними формами) з аналізом отриманих результатів.
9. Інструкція користувачу, в якій міститься пояснення щодо правил користування програмою.
10. Висновки.
11. Список використаної літератури.

Курсова робота оформляється на листах формату A4, що зверху нумеруються і зшиваються з лівої сторони. Ліве поле повинне бути від 25 до

30 мм, праве - не менше 10 мм; верхнє – 20 мм; нижнє – 20 мм. **Номер сторінки ставлять у верхньому правому куті. Першою сторінкою курсової роботи є титульний лист, але номер на ньому не ставиться.**

Текст повинен відповідати змісту. Всі пункти змісту викладаються чітко і коротко.

## **2.2. Вимоги до розробки програми**

Програма для реалізації курсової роботи розробляється на мовах програмування *C/C++*, *C#*, *Python*, *Java Script* з використанням середовищ розробки *Borland Delphi*, *Microsoft Visual Studio* тощо.

При написанні програми потрібно дотримуватися наступних вимог:

- початкові дані та, можливо, результати обчислень зберігати у типованих файлах;
- програма повинна містити зручний для користування інтерфейс;
- тестування програми провести на прикладі, що дозволяє перевірити всі режими роботи програми.

## **2.3. Основні етапи виконання курсової роботи**

Виконання курсової роботи складається з наступних етапів:

**1. Змістовна постановка задачі.** Після отримання завдання для виконання курсової роботи студент повинен вивчити предметну область що пов'язана з темою курсової роботи та її застосуванням. Для цього він самостійно підбирає і опрацьовує необхідну літературу по тематиці роботи, консультуючись при цьому з науковим керівником.

**2. Аналіз поставленої задачі.** Перш ніж приступати до програмної реалізації розроблених алгоритмів потрібно ретельно проаналізувати задачу. Якщо практична реалізація алгоритму розв'язання задачі досить складна, він може бути спрощений за погодженням з керівником.

**3. Розробка і опис алгоритму розв'язання задачі.** Використовуються ідеї об'єктно-орієнтованого аналізу. Формуються основні класи та об'єкти, що братимуть участь у вирішенні поставленої задачі. Будуються діаграми UML. Вибираються необхідні шаблони проектування.

**4. Розробка і написання програми розв'язання задачі.** При написанні програми потрібно користуватися сучасними засобами та розробками в області програмування.

# **3. ЗАХИСТ КУРСОВОЇ РОБОТИ**

## **3.1. Оформлення роботи**

Оформлена курсова робота подається студентом до захисту разом з текстом програми на цифровому носії (CD/DVD) або опублікована у відкритому репозиторії GitHub. Захист роботи проходить у формі співбесіди з демонстрацією слайдів презентації на проекторі.

До захисту студент повинен підготувати презентацію, яка повністю розкриватиме зміст та етапи виконання курсової роботи. Презентацію доцільно робити використовуючи програмні засоби Microsoft PowerPoint.

Під час захисту потрібно коротко розповісти про зміст задачі, основні етапи її розв'язання, аргументувати вибір того чи іншого методу розв'язання задачі, пояснити як складались алгоритми і відповідна їх програмна реалізація. На поставлені запитання по виконаній роботі студент повинен дати вичерпні відповіді. Мета опитування полягає у встановленні глибини засвоєння студентом відповідного матеріалу з курсової роботи, ступеня самостійності її виконання. При оцінюванні роботи враховується якість її виконання і оформлення, своєчасність виконання етапів роботи, результати співбесіди.

### **3.2. Методика оцінювання курсових робіт**

Оцінювання курсових проектів проводиться за 100-бальною шкалою, відповідно до розділу.

При цьому оцінка виставляється за такими критеріями:

#### **1. Дотримання графіка розробки курсового проекту (5 балів).**

Студент може отримати максимальний бал згідно цього критерію, якщо він представляв усі матеріали курсового проекту для контролю викладачу відповідно до затвердженого графіка, а також вчасно завершив виконання проекту.

**2. Оформлення пояснювальної записки (до 10 балів).** Максимальна оцінка по цьому критерію виставляється тоді, коли пояснювальна записка відповідає усім вимогам, встановленим у методичних вказівках до курсової роботи, а також коли записка оформлена акуратно і згідно стандартів. Особлива увага при цьому приділяється оформленню малюнків і таблиць. Також бали знімаються тоді, коли у записці недостатньо розкрито зміст певних розділів курсового проекту.

**3. Інтерфейс та функціональність програми (до 10 балів).** Студент отримує максимальний бал у тому випадку, коли розроблена ним програма має забезпечує користувача усіма потрібними йому функціями, має зручний інтерфейс, а також забезпечує коректне формування і виведення результатів своєї роботи. Висновок про кількість балів, отриманих студентом відповідно до цього критерію, робиться комісією на основі демонстрації роботи програми та її тестування, яке проводиться під час доповіді.

**4. Індивідуально-пошукова робота студента (до 15 балів).** Бали за проведення індивідуально-пошукової роботи нараховуються студентам за:

- дослідження предметної області, для якої буде створюватись програма або база даних;
- обґрунтування вибору оптимального методу введення вхідних даних, а також виводу результатів роботи програми на основі кількох варіантів;

- дослідження можливостей виконати поставлену задачу іншими (альтернативними) методами;

- проведення тестування роботи програми з використанням різних методів і підходів, що дозволяють швидко оцінити правильність її роботи. При цьому може враховуватись заповнення вхідних даних випадковими числами або складання спеціалізованих програм-додатків для заповнення баз даних або структур відповідною інформацією. Порядок роботи цих додатків повинен бути описаний у розділі 6.

- дослідження методики практичного застосування розробленої програми. При цьому інформація про практичне використання програми повинна бути представлена у висновках, а також у розділі 6 під час формування тестових наборів даних, які повинні відповідати реальним даним, що будуть використовуватись під час експлуатації розробленої програми.

При цьому посилання на конкретні дії, які виконував студент в рамках пошукової роботи повинні бути вказані у висновках, а також коротко висвітлені студентом під час доповіді на захисті курсових проєктів.

### **3.3. Оцінки за захист курсової роботи**

**1. Доповідь студента під час захисту проєкту (до 30 балів).** Даний критерій є основним при оцінюванні курсових проєктів. Він визначає здатність студента вільно виступати перед аудиторією, доповідаючи про процес розробки та особливості роботи створеної ним програми, користуючись набутими у коледжі знаннями із різних дисциплін. Комісією в першу чергу оцінюється ґрунтовність та лаконічність відповіді, послідовність викладення матеріалу, вміння оперувати основними термінами та поняттями. Максимальний бал студент отримує, якщо під час доповіді студент вільно володіє термінологією, орієнтується у всіх деталях роботи програми, а також у предметній області, для якої ця програма розробляється.

**2. Відповіді студента на додаткові запитання комісії (до 20 балів).** Студент отримує максимальний бал по цьому критерію, якщо він надає вичерпні та точні відповіді на додаткові запитання комісії, які стосуються розробленого ним продукту, основних термінів мови програмування або системи керування базами даних, які використовувались для розробки програми, а також основних понять та термінів із дисциплін та галузей науки, до яких належить предметна область розробленої студентом програми.

**3. Участь студента у захисті робіт інших студентів (до 10 балів).** Студенти отримують бали за цим критерієм тоді, коли під час захисту курсових проєктів іншими студентами вони задають їм додаткові запитання, а також вносять свої пропозиції щодо вдосконалення розроблених програм. За кожне правильно та обґрунтовано задане запитання студент може отримати від 1 до 2 балів.

## 4. ОРІЄНТОВНИЙ ПЕРЕЛІК ТЕМ КУРСОВОЇ РОБОТИ З ДИСЦИПЛІНИ «ПРОГРАМУВАННЯ»

### 4.1. Жуковський В.В. - 121 «Інженерія програмного забезпечення»:

1. Розробка мобільного додатку з використанням технології Xamarin.
2. Реалізація алгоритмів архівування файлів.
3. Програмний аналіз супутникових знімків на предмет NDVI індексу рослинності.
4. Програмна реалізація бази даних агрохіміпоказників.
5. Демонстраційно-навчальна програма алгоритмів теорії чисел.
6. Дизасемблювання ехе-файлів, що написані на мовах C++ та Pascal.
7. Дослідження алгоритмів роботи з деревами (червоно-чорні, скошені, AVL дерева). Алгоритми обходу дерева в ширину та глибину.
8. Інформаційно-довідкова система "Облік роботи приватних таксобусів".
9. Інформаційно-довідковий портал навчальної літератури.
10. Реалізація механізму надсилання SMS за допомогою Viber.
11. Інформаційна система обліку відвідування студентами заходів.
12. Програмна оптимізація системи роботи світлофорів в умовах міста.
13. Програма для підрахунку матеріалів для будівництва забору.
14. Демонстраційна програма аналізу предсталення графів (об'єкти і вказівники, матриця, послідовність зв'язків). Алгоритми проходження графа в ширину та глибину.
15. Розробка системи «Smart Postbox» (поштовий ящик із сигналізатором чи є там лист чи немає).
16. Інформаційна система для роботи з генеалогічним деревом родоводу.
17. Програма зберігання карти місцевості у пам'яті комп'ютера.
18. Програма пошуку найкоротшого виходу з лабіринту.
19. Програма пошуку оптимального варіанту купівлі нерухомості.
20. Програма реалізація сервісу перегляду віддаленого робочого столу.
21. Програма-симулятор сонячної системи.
22. Програмна реалізація алгоритмів з використанням динамічних структур даних.
23. Програмна реалізація гри "Точки".
24. Програмна реалізація додатку до Для захисника Вітчизни.
25. Програмна реалізація додатку до Дня Святого Валентина.
26. Програмна реалізація додатку розв'язання японського кросворду.
27. Програмна реалізація додатку розпізнавання OR-кодів.
28. Програмна реалізація додатку створення OR-кодів.
29. Програмна реалізація інформаційної системи кафедри.
30. Програмна реалізація калькулятора підрахунку матеріалів для монтажу стелі з гіпсокартону.
31. Програмна реалізація машини Тюрінга.



32. Програмна реалізація мурашиного алгоритму для оптимізації маршруту.
33. Програмна реалізація оболонки для демонстрації роботи експлоїтів.
34. Програмна реалізація сервісу перевірки на плагіат.
35. Програмна реалізація системи роботи з rainbow таблицями.
36. Програмна реалізація стрічкового компілятора.
37. Програмна реалізація трьохвимірного лабіринту.
38. Програмна реалізація управління космічним апаратом на орбіті і при польоті на Місяць.
39. Програмна реалізація утиліти для наповнення тестами системи Moodle.
40. Розробка програмного забезпечення для захисту від копіювання.

#### **4.2. Прищепя О.В. - 113 «Прикладна математика»:**

1. Демонстраційно-навчальна програма алгоритмів теорії графів.
2. Демонстраційно-навчальна програма для навчальної дисципліни "Дискретна математика" (розділи теорія множин і відношень / теорія графів / комбінаторика).
3. Довідкова система для пасажирів громадського транспорту (реалізувати можливість пошуку варіантів руху з мінімальною кількістю пересадок).
4. Моделювання роботи системи опалювання в залежності від прогнозу погоди (наприклад, вважається, що система опалювання є інерційною і програма буде автоматично отримувати прогноз погоди з інтернет сайту та коригувати роботу котла з затримкою).
5. Програмна реалізація гри "Імітація роботи фондової біржі" (задається можлива кількість фірм, які пропонують для продажу власні акції).
6. Програмна реалізація гри "Формування оптимального пакету цінних паперів". Використовуються математичні моделі. Традиційні і нетрадиційні підходи.
7. Програмна реалізація додатку "Re-Write" (програма, що дозволить змінювати заданий текст. Замінює слова синонімами "і т.д." на "і т.п." без втрати змісту).
8. Програмна реалізація додатку до Дня Програміста.
9. Програмна реалізація додатку під мобільну платформу Android (можна використовувати програми-емулятори при відсутності пристроїв).
10. Програмна реалізація додатку під мобільну платформу Windows (можна використовувати програми-емулятори при відсутності пристроїв).
11. Програмна реалізація додатку під платформу iPhone (для прикладу програма, що запам'ятовує пінкоди, паролі іт.д.).

12. Програмна реалізація чат-бота (може бути декілька тем, боти на різні теми). При виборі теми необхідно уточнити тему (що буде робити бот).
13. Розробка програмного забезпечення для визначення програми-клона (пошук по сервісам вихідних кодів і т.д. і т.п.).
14. Розробка програмного забезпечення для дослідження геометричних властивостей фракталів.
15. Розробка програмного забезпечення для знаходження "реферату-клона" вважається, що є база рефератів або курсових.
16. Розробка програмного забезпечення для планування розважальної програми.
17. Розробка програмного забезпечення для пошуку оптимальний матеріалів при теплоізоляції будинку.
18. Розробка програмного забезпечення для пошуку оптимального розпилу дерева.
19. Розробка програмного забезпечення для пошуку оптимального розпилу фанери ДСП при виготовленні меблів.
20. Розробка програмного забезпечення для прийому замовлень служби таксі.
21. Розробка програмного забезпечення для проведення квестів.
22. Розробка програмного забезпечення для роботи з системою Cortana.
23. Розробка програмного забезпечення для розв'язання нелінійних рівнянь.
24. Розробка програмного забезпечення для створення головоломок "Word Search" для дітей.
25. Розробка програмного забезпечення комп'ютерного моделювання розв'язку задач лінійного програмування.
26. Розробка програмного забезпечення наближеного обчислення визначених інтегралів (звичайних, подвійних, потрійних).
27. Розробка програмного модуля для побудови графіків в трьохвимірному просторі.
28. Розробка програмного продукту з вибору оптимальної сівозміни.
29. Розробка програмного продукту з використанням фреймворку Xamarin.
30. Розробка системи SMS оповіщення батьків про успішність їхніх дітей.
31. Розробка системи моніторингу вступної компанії.
32. Розробка системи моніторингу стану доріг.
33. Сівозміна на основі попередніх даних.

#### 4.3. Ярошак С.В. - 122 «Комп'ютерні науки»:

1. Arduino.
2. MISRA та особливості написання безпечного коду для мікроконтролерів.
3. Raspberry Pi.
4. Алгоритм AlphaZero
5. Аналіз даних з використанням бібліотеки Pandas.
6. Апаратні атаки на мікроконтролери.
7. Асиметрична криптографія. Шифр RSA.
8. Використання Angular.
9. Використання D3.js
10. Використання Docker.
11. Використання Ember.js
12. Використання Microsoft Azure.
13. Використання Node.js
14. Використання opencv в комп'ютерному баченні.
15. Використання React.js.
16. Використання TensorWlow.js при машинному навчанні на javascript.
17. Використання Vue.js
18. Використання архітектури transformer в NLP.
19. Використання бібліотек MLlib і PySpark при класифікації текстів.
20. Використання бібліотеки asuncio для асинхронного програмування в Python.
21. Використання бібліотеки Pygame розробки відеоігор.
22. Використання бібліотеки машинного навчання XGBoost.
23. Використання інструментів GitBlame, GitBisect, GitGrep при відлагодженні програм.
24. Використання мови R в аналізі даних.
25. Використання хмарного сервісу Heroku для розгортання web-додатків.
26. Емулятори мікроконтролерів.
27. Згорткові нейронні мережі.
28. Контролер ShIoTiny.
29. Криптоаналіз Виженера. Метод індекса співпадань та автокореляційний метод.
30. Мова програмування Dart.
31. Особливості роботи з мікроконтролером ESP32 та його файловою системою SPIFFS.
32. Рекурентні нейронні мережі.
33. Рефлексія в C#.
34. Робота з метекласами в Python.
35. Сімейство мікроконтролерів ARM та особливості їх програмування.
36. Сімейство мікроконтролерів AVR та особливості їх програмування.

37. Сімейство мікроконтролерів CORTEX та особливості їх програмування.
38. Сімейство мікроконтролерів MCS 51 та особливості їх програмування.
39. Сімейство мікроконтролерів PIC та особливості їх програмування.
40. Сімейство мікроконтролерів STM та особливості їх програмування.

## **5. ПРИКЛАД ОФОРМЛЕННЯ БІБЛІОГРАФІЧНОГО ОПИСУ**

Список використаної літератури розміщують одним із таких способів: у порядку появи посилань у тексті, в алфавітному порядку прізвищ перших авторів або заголовків, в хронологічному порядку.

Зразок оформлення бібліографічного опису.

### **5.1. Підручника, навчального посібника (один, два або більше авторів):**

1. Гради Буч, Джеймс Рамбо, Ивар Якобсон Введение в UML от создателей языка: ДМК Пресс, 2011 г., 496 с.
2. Гради Буч, Роберт А. Максимчук, Майкл У. Энгл, Бобби Дж. Янг, Джим Коналлен, Келли А. Хьюстон Объектно-ориентированный анализ и проектирование с примерами приложений: Вильямс, 2010 г., 720 с.
3. Диомидис Спинеллис, Георгиос Гусиос Идеальная архитектура. Ведущие специалисты о красоте программных архитектур: Символ-Плюс, 2010 г., 528 с.
4. Майк Кон Scrum. Гибкая разработка ПО: Вильямс, 2011 г., 576 с.
5. Эрик Эванс Предметно-ориентированное проектирование (DDD). Структуризация сложных программных систем: 2010 г., 448 с.
6. Мартин Фаулер Рефакторинг. Улучшение существующего кода: Символ-Плюс, 2008 г., 432 с.
7. Роберт Мартин Чистый код. Создание, анализ и рефакторинг: Питер, 2010 г., 464 с.
8. Roman Pichler Agile Product Management with Scrum: Creating Products that Customers Love: Addison-Wesley Professional, 2010 г., 160 с.

### **5.2. Збірника наукових праць:**

Вісник. Економіка. Сучасні методи управління підприємством. Зб. наук. праць. Вип.4(22).Ч.1. – Рівне: НУВГП, 2004. – 254с.

### **5.3. Статті з журналу:**

Захарін С.В. Кредитування інвестиційної діяльності // Фінанси України. — 2004. — №4. — с.97-105.

#### **5.4. Нормативних документів:**

Земельний кодекс України: Закон України від 13 березня 1992р. №2196 — XII // Відомості Верховної Ради України. – 1992. – №23. – ст.354. – с.743–786.

### **6. РЕКОМЕНДОВАНА ЛІТЕРАТУРА**

#### **6.1. Основна література**

1. Banks A., Porcello E. Learning React: Functional Web Development with React and Redux. URL: <https://it.b-ok2.org/book/2885521/e24cfb> (дата звернення 20.03.2020).
2. Deitel P., Deitel H. Python for Programmers: with Big Data and Artificial Intelligence Case Studies. URL: <https://it.b-ok2.org/book/4975489/3f6f61?dsource=recommend> (дата звернення 10.03.2019).
3. Dow C. Internet of Things Programming Projects: Build modern IoT solutions with the Raspberry Pi 3 and Python. URL: <https://it.b-ok2.org/book/3695529/6b4930?dsource=recommend> (дата звернення 20.03.2020).
4. Huber T. Getting Started with TypeScript : Includes Introduction to Angular. URL: <https://it.b-ok2.org/book/2863272/8d8611> (дата звернення 10.03.2019).
5. Johansson R. Numerical Python: Scientific Computing and Data Science Applications with Numpy, SciPy and Matplotlib.
6. Matthes E. Python Crash Course: A Hands-On, Project-Based Introduction to Programming. URL: <https://it.b-ok2.org/book/2708675/f4a3d9?dsource=recommend> (дата звернення 10.03.2020).
7. Svetlin Nakov & Co. Fundamentals of Computer Programming with C#. Sofia, 2013. P. 1122. URL: [www.introprogramming](http://www.introprogramming) (дата звернення 20.03.2020).
8. URL: <https://it.b-ok2.org/book/3649126/4409f0?dsource=recommend> (дата звернення 20.02.2020).
9. Zammetti F. Practical React Native: Build Two Full Projects and One Full Game using React Native. URL: <https://it.b-ok2.org/book/3625108/eccaac> (дата звернення 10.03.2019).
10. Шилдт Г. C# 4.0: полное руководство. ООО "И.Д. Вильямс", 2011. С. 1465.
11. Шилдт Г. Полный справочник по C#. М.: Вильямс, 2004. С. 752.

#### **6.2. Електронні ресурси**

1. Цифровий репозиторій НУБГП. URL: <http://ep3.nuwm.edu.ua> (дата звернення: 31.02.2020).
2. Веб-сервіс для спільної розробки програмного забезпечення. URL: <https://github.com/> (дата звернення: 31.02.2020).
3. Офіційний сайт спільноти розробників UML. URL: <https://www.omg.org/> (дата звернення: 31.02.2020).

## **ДОДАТОК**

Приклад оформлення титульного листа

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВОДНОГО ГОСПОДАРСТВА ТА  
ПРИРОДОКОРИСТУВАННЯ**

**Кафедра комп'ютерних наук та прикладної математики**

### **КУРСОВА РОБОТА**

з дисципліни "Програмування"  
на тему

---

(назва теми роботи)

Виконав студент ННІАКОТ:

група \_\_\_\_\_

\_\_\_\_\_  
(прізвище, ім'я, по батькові)

Керівник:

\_\_\_\_\_  
(науковий ступінь, вчене звання,  
прізвище, ініціали)

Рівне - 20\_\_